Lab

Daniel Andronov

Tuesday 11th October, 2016

3.0 Prelab

Question 1)

a) False, each object needs its own request.

b) True, if using HTTP 1.1, the connection will not be closed until the client says so.

c) False, after each response, the connection is closed.

d) False, that field indicates when the packet was sent.

Question 2)

a) The end-to-end delay would be $2N + 2$ RTT, where RTT is the round trip time. Two RTT's for setting up the connection and getting the base

b) If using HTTP 1.1, the end-to-end delay woudl be $N + 2$ RTT, 2 RTT for setting up the connection and the base HTML file, and N RTT for each subsequent object.

c) If usine HTTP 1.0 & four parallel connections, then the end-to-end delay would be $2N/4$ or $N/2$.

Question 3) The most modern iteration of FTP is RFC 959. It's well known port numbers are 20 & 21. FTP has two port numbers assigned to it because it was found the the useing the header to communicate commands and manage directories created too much latency. One connection is used to send commands and the other to move data.

Question 4) From the figure below, the `dig` command returned nine answers but ranked `relay.bu.edu` above the others. A secondary `dig` revealed that `dig` has several IP addresses, most like a redundacy safety measure, but the first was `128.197.228.27`.

3.1 HTTP

Question 1) The browser is running HTTP1.1.
My computer's IP address is 172.16.160.130

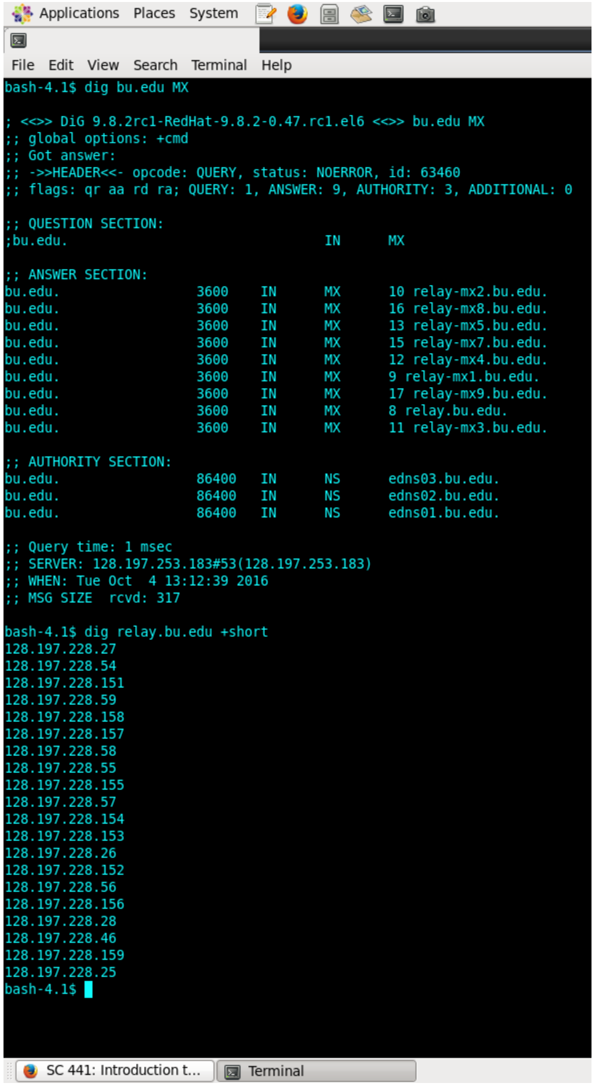
Question 2) bc.edu's IP address is 136.167.2.220

Question 3) The source port is 54910 and the destination port is 80.

Question 4) The source is now 80 and the destination is 54910.

Question 5) The server is running Apache.
The HTML file the was requested was last modified on Tuesday, October 4th 2016 at 17:17:01 GMT.

Figure 1: The results of the `dig` queries.



```
bash-4.1$ dig bu.edu MX

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.47.rc1.el6 <<>> bu.edu MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63460
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;bu.edu.                                IN      MX

;; ANSWER SECTION:
bu.edu.                 3600    IN      MX      10 relay-mx2.bu.edu.
bu.edu.                 3600    IN      MX      16 relay-mx8.bu.edu.
bu.edu.                 3600    IN      MX      13 relay-mx5.bu.edu.
bu.edu.                 3600    IN      MX      15 relay-mx7.bu.edu.
bu.edu.                 3600    IN      MX      12 relay-mx4.bu.edu.
bu.edu.                 3600    IN      MX      9 relay-mx1.bu.edu.
bu.edu.                 3600    IN      MX      17 relay-mx9.bu.edu.
bu.edu.                 3600    IN      MX      8 relay.bu.edu.
bu.edu.                 3600    IN      MX      11 relay-mx3.bu.edu.

;; AUTHORITY SECTION:
bu.edu.                 86400   IN      NS      edns03.bu.edu.
bu.edu.                 86400   IN      NS      edns02.bu.edu.
bu.edu.                 86400   IN      NS      edns01.bu.edu.

;; Query time: 1 msec
;; SERVER: 128.197.253.183#53(128.197.253.183)
;; WHEN: Tue Oct  4 13:12:39 2016
;; MSG SIZE  rcvd: 317

bash-4.1$ dig relay.bu.edu +short
128.197.228.27
128.197.228.54
128.197.228.151
128.197.228.59
128.197.228.158
128.197.228.157
128.197.228.58
128.197.228.55
128.197.228.155
128.197.228.57
128.197.228.154
128.197.228.153
128.197.228.26
128.197.228.152
128.197.228.56
128.197.228.156
128.197.228.28
128.197.228.46
128.197.228.159
128.197.228.25
bash-4.1$
```

Question 6) There were a total of 9 GET messages sent. This could be either for reduancy, so that if packets are lost, at least one will make it through the network to bc.edu in order to initiate a respone, or part of some parallelization of certain process that each send their own HTTP GET message.

3.2 DNS

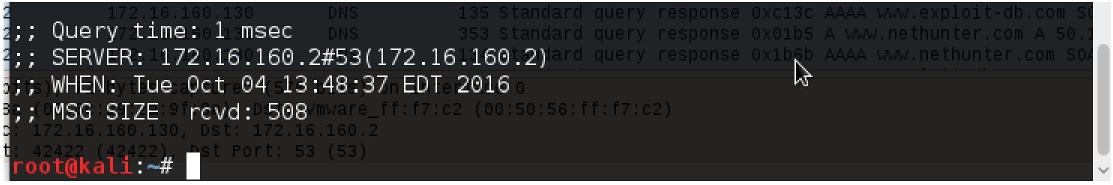
Question 7) The query and response are both send via UDP.

Question 8) The source port is 42422 and the destination port is 53.

Question 9) The DNS query is sent to 172.16.160.2 and `dig` reveals that the DNS local server IP is the same

Question 10) The DNS query message is of type A, and contains no answers or records

Figure 2: The result of the dig command reports the local DNS IP



```
;; Query time: 1 msec
;; SERVER: 172.16.160.2#53(172.16.160.2)
;; WHEN: Tue Oct 04 13:48:37 EDT 2016
;; MSG SIZE  rcvd: 508

root@kali:~# 
```

Question 11) The DNS response message returns with 1 answer, 5 authority, and 8 additional records, for a total of 13 records. The first answer is the name and IP address of the desired name server as it is known in the cache. The authoratative records contain the names and IP address of authoratative name servers for the desired domain that contain the most current information. The additional records contain the TTL's of the authoratative records, as well as their names and IP addressess, as stored in the cache.

Question 12) The maxiumum TTL of a DNS record in 86400 seconds or 3 years as the TTL number is a 32 bit value.

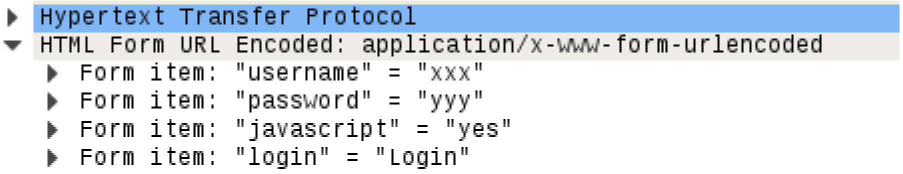
Question 13) Type "AAAA" queries request the IPv6 128 bit address instead of the regular 32 bit IPv4 address. It is defined by RFC 3596.

3.3 Authentication Mechanisms and HTTP

Question 14) The HTTP method POST was used to send the login information.

Question 15) The password was submitted in plain text, as per the below figure. This is a fairly serious security concern.

Figure 3: This is the contents of the POST HTTP message that was used to send the login information



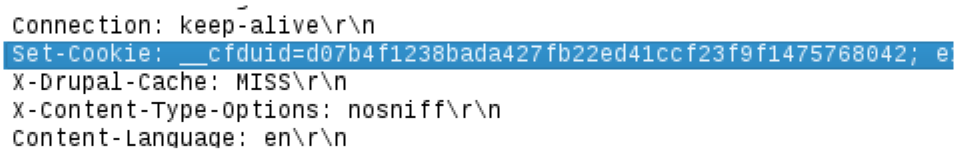
```
▶ Hypertext Transfer Protocol
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
   ▶ Form item: "username" = "xxx"
   ▶ Form item: "password" = "yyy"
   ▶ Form item: "javascript" = "yes"
   ▶ Form item: "login" = "Login"
```

3.4 Cookies

Question 16) In the first HTTP GET message, there is no cookie ID as it was deleted when the history of the browser was cleared.

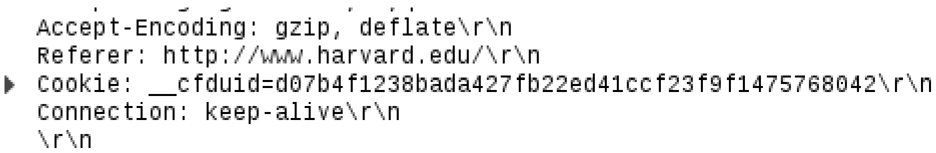
Question 17) The first HTTP response message sets the cookie ID to `d07b4f1238bada427fb22ed41ccf23f9f1475768042`.

Figure 4: The cookie ID provided in the response message the initial GET message



```
Connection: keep-alive\r\n
Set-Cookie: __cfduid=d07b4f1238bada427fb22ed41ccf23f9f1475768042; e
X-Drupal-Cache: MISS\r\n
X-Content-Type-Options: nosniff\r\n
Content-Language: en\r\n
```

Figure 5: This figure shows that the cookie id in the subsequent HTTP GET message matches that which was given in the previous response



```
  Accept-Encoding: gzip, deflate\r\n
  Referer: http://www.harvard.edu/\r\n
▶ Cookie: __cfduid=d07b4f1238bada427fb22ed41ccf23f9f1475768042\r\n
  Connection: keep-alive\r\n
  \r\n
```

Question 18) In the second HTTP GET message, the cookie ID from the previous response is that which is included in the message.